(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number : **0 529 933 A2**

(12) **EUROPEAN PATENT APPLICATION**

(21) Application number : **92307587.3**

(51) Int. Cl.[5] : **G09G 3/36**

(22) Date of filing : **19.08.92**

(30) Priority : **23.08.91 US 749072**

(43) Date of publication of application : **03.03.93 Bulletin 93/09**

(84) Designated Contracting States : **DE FR GB**

(71) Applicant : **MOTOROLA, INC.**
1303 East Algonquin Road
Schaumburg, IL 60196 (US)

(72) Inventor : **Wong, Harvey**
1ST, 26 Sheung Fung Street, Fung Wong Village
Wong Tai Sin, Kowloon (HK)

(74) Representative : **Hudson, Peter David et al**
MOTOROLA European Intellectual Property Operations Jays Close Viables Industrial Estate
Basingstoke Hants RG22 4PD (GB)

(54) **Data stream altering system, for example in a LCD driver.**

(57) A LCD controller/driver utilizes a bi-directional data ring to enable quick and easy data alteration and shifting. The controller/driver couples cascaded segment drivers (32, 34) to a row data serial interface (24) which links the drivers with a control circuit. The cascaded segment drivers coupled to the row data serial interface define the bi-directional data ring. Display data input into the segment drivers may be altered by shifting the data into the data ring and transmitting such data to, or through, other segment drivers and to the row data serial interface, where the data is altered by inputs from the control circuit. The control circuit is further coupled to a horizontal pointer register (64). The horizontal point register has latched, from instructions from the MPU, two locations within the data stream of the data ring. These pointers define the boundaries of the area (active region) to be amended or replaced by the MPU.

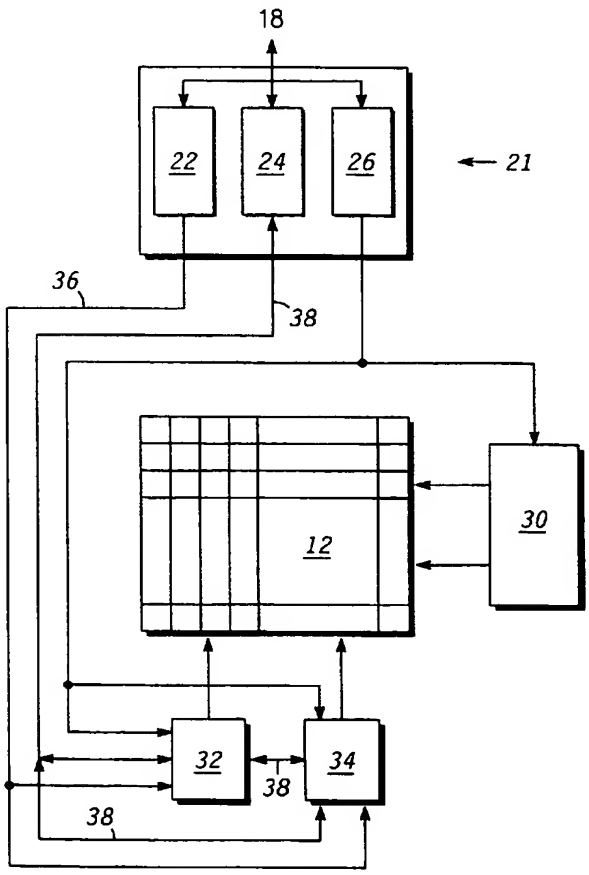


FIG. 2

EP 0 529 933 A2

Background of the Invention

This invention relates, in general, to a graphical support unit for liquid crystal displays (LCD).

Traditionally, good graphic support equated to complicated LCD controllers. As displayed graphics on LCD screens become more and more complex, conventional support devices increased in number. As graphics manipulation complexities increase the size of the graphics driving devices, the drain on stored power also increases. Power drain is an obvious problem when the LCD is part of a hand held electronics device driven by power stored in batteries. Furthermore, size of the LCD support devices must be carefully considered when designing hand-held units which are constantly decreasing in size.

An additional trend in the electronics market is increasing the speed of the device and increasing the device's capability. The capability of the device is often dependent upon the load on the micro-processing unit (MPU). If high speed support elements can perform many of the functions normally required of the MPU, the MPU will be freed up to perform more advanced functions.

Summary of the Invention

According to the present invention, a LCD controller/driver utilizes a bi-directional data ring to enable quick and easy data alteration and shifting. The controller/driver couples cascaded segment drivers to a row data serial interface which links the drivers with a control circuit. The cascaded segment drivers coupled to the row data serial interface define the bi-directional data ring. Display data input into the segment drivers may be altered by shifting the data into the data ring and transmitting such data to, or through, other segment drivers and to the row data serial interface, where the data is altered by inputs from the control circuit. The control circuit is further coupled to a horizontal pointer register. The horizontal point register has latched, from instructions from the MPU, two locations within the data stream of the data ring. These pointers define the boundaries of the area (active region) to be amended or replaced by the MPU.

The above and other features and advantages of the present invention will be better understood from the following detailed description taken in conjunction with the accompanying drawings.

Brief Description of the Drawing

Fig. 1 is a generalized schematic of a LCD driver and control unit according to the present invention.

Fig. 2 shows the schematic of Fig. 1 broken into components.

Fig. 3 shows the elements of the segment driver of Fig. 2.

Fig. 4 is a schematic of a LCD controller/driver with a split screen LCD according to the present invention.

Fig. 5 shows the data transfer between screens on a split screen LCD according to the present invention.

Fig. 6 is a schematic of the control circuit of Fig. 2 according to the present invention.

Fig. 7 is a schematic of a sequential write operation according to the present invention.

Detailed Description of the Invention

Fig. 1 shows a LCD controller/driver 10 coupled to a LCD 12. LCD controller/driver 10 comprises a segment control interface 22, a row data serial interface 24, a LCD timing generator 26, and LCD drivers 28. Segment control interface 22, row data serial interface 24, and LCD timing generator 26, which generates the timing signals, are shown as part of a micro-control unit (MCU) 21.

Control circuit 20 is coupled to a micro-processing unit (MPU) 18 which is not shown but indicated by number in Fig. 1. Since controller/driver 10 is used for LCD graphic control, information is transferred back and forth between interfaces 22, 24, and 26 and MPU 18.

MPU 18 sends timing signals to LCD timing generator 26, and sends segment control signals to interface 22. Information may be transmitted both ways between MPU 18 and row data serial interface 24.

Information is relayed both ways between LCD drivers 28 and row data serial interface 24.

Row data serial interface 24 is a serial interface which supports high speed serial bits transfer among LCD drivers 28 and MPU 18, pointer functions, variable pitch selection, screen width definition, and other serial data handling functions. As will be further described later, row data serial interface 24 and cascaded segment drivers are connected to form a data ring where the data flow in the data ring is bi-directional. Rows, or portions thereof, of display data are transferred back and forth between MPU 18 and drivers 28 through row data serial interface 24. This transfer of information allows ready access to the data for updates or alterations. All data for display is transferred through row data serial interface 24.

Segment control interface 22 supplies commands to drivers 28. The correlation between segment control interface 22 and row data serial interface 24 will be described later in conjunction with Fig 3.

LCD timing generator 26 supplies timing signals for display synchronization among the individual drivers of drivers 28.

Fig. 2 shows the circuit of Fig. 1 having the elements of LCD drivers 28 broken out. LCD drivers 28 comprises backplane (or row) driver(s) 30 (shown as

a single driver for description purposes in Fig. 2) and segment (or column) drivers 32 and 34. Although only two segment drivers are shown in Fig. 2, it should be understood that any number of segment drivers may be coupled in series similar to drivers 32 and 34, depending upon the width of LCD 12.

Backplane driver 30 is preferably a CMOS chip consisting of 80 high voltage LCD driving signals. Backplane driver 30 supports a multiplex ratio of 32 to 256. Backplane driver 30 can be cascaded in multiples to achieve a higher driver count than 65. One having skill in the art will recognize that backplane driver 30 is the vertical display control for LCD 12. Backplane driver 30 activates the row of LCD 12 into which a row of display data is to be displayed.

Backplane driver 30 is coupled to LCD timing generator 26.

Segment drivers 32 and 34 are both coupled to Segment control interface 22, LCD timing generator 26, and bi-directionally to row data serial interface 24. Furthermore, segment drivers 32 and 34 are coupled bi-directionally to each other.

The data ring mentioned in relation to Fig. 1 can be readily seen in Fig. 2. Display data is input from row data serial interface 24 to segment drivers 32 and 34 through a one-bit bi-directional serial line (38). Signal lines 36 from segment control interface 22 input control signals to segment drivers 32 and 34. The display data can be shifted from one segment driver to another, updated by MPU 18, or displayed on LCD 12.

Fig. 3 shows the elements of segment driver 32, as well as the relation between segment driver 32 and segment control interface 22 and the other elements of data ring 38.

Segment driver 32 generally comprises timing logic 40 coupled to LCD timing generator 26, control logic 42 coupled to segment control interface 22 via 8-bit bus lines 36, high voltage drivers array 44 coupled to LCD 12, display data latch array 46 coupled to timing logic 40 and high voltage drivers array 44, and random access memory (RAM) 48 coupled to display data latch array 46. RAM 48 is further coupled to timing logic 40, and to control logic 42. Segment driver 32 also comprises shift register 50 which is coupled to control logic 42, RAM 48, and linked within the data ring 38.

RAM 48 is N rows high by 160 bits wide. Similarly, shift register 50, display data latch array 46, and high voltage drivers array 44 are 160 bits wide. When two segment drivers are cascaded as with segment drivers 32 and 34, LCD 12 may be 320 bits wide. The present invention does not contemplate specifically limiting the number of cascaded segment drivers within the data ring to two. The limit of number of segment drivers is dependent upon the size of horizontal loop size register (described subsequently). The present embodiment, a 10-bit loop size, is used allowing up to six segment drivers to be cascaded.

Display data is received by shift register 50 from the data ring 38. Shift register 50 may input the display data into RAM 48 or shift it on through data ring 38. The transfer of the display data from shift register 50 to RAM 48, is controlled by the control signal from segment control interface 22 as indicated by line 52. As soon as display data in data ring 38 is in its designated position, MPU will direct all segment drivers, through control logic 42, to fetch the display data from shift register 50 to RAM 48.

The control signal from control logic 42 also governs the row of data shifted from RAM 8 to display data drivers array 46 in conjunction with timing signals from timing logic 40. Data latched in display data drivers array 46 will be clocked into high voltage drivers array 44, and subsequently displayed on LCD 12.

Just as the MPU can direct control logic 42 to fetch display data into RAM 48 of the segment drivers, the MPU can also direct control logic 42 to retrieve display data from RAM 48 into shift register 50. This brings display data back into the data ring. Display data which is retrieved into data ring 38 may be amended, replaced, or transferred to another screen.

LCD controller/driver 10 may be coupled to a split panel LCD incorporating LCD 12 and 13 as shown in Fig. 4. Fig. 4 shows a split panel having LCDs 12 and 13 coupled to segment drivers 32/34 and 33/35 respectively. LCDs 12 and 13 are both coupled to backplane driver 30. Segment drivers 32/34 and 33/35 are coupled to MCU 21. In this embodiment, two loops make up the data ring. One loop operates with segment drivers 32 and 34, while the other loop operates with segment drivers 33 and 35. Generally, each loop operates independently of the other. However, both loops may be connected into a single data ring as shown in Fig. 5 by path switch and combinational logic (PCL) 60. Since data to be displayed is already stored in the RAM of the segment driver, if there is a means of transferring data between different banks of the segment driver, MPU 18's supervision can be kept at a minimum. MPU 18 can cause the data from the RAM of the segment drivers for one screen to be fetched into the data ring. MPU 18 then has PCL 60 couple the loops for upper and lower screens together and the data is transferred directly from the segment drivers of one screen to the segment drivers of the other screen. Although Fig. 5 shows a unidirectional data flow, it should be recognized that the data ring is bi-directional, or the data can be transferred both ways. Further, data can be injected into the bit stream as the data is being transferred from one screen to the other. This method of transferring data from screen to screen is faster than conventional methods and reduces the burden on the MPU.

The control of display information is understood by referencing Fig. 6. A control circuit 20 is coupled to a horizontal loop size register 62, a horizontal pointer register 64, a horizontal pitch register 66, a mode and

attribute register 68, and a status register 70. Control circuit 20 receives signals from all registers except status register 70 where control circuit 20 inputs status signals to status register 70. Control circuit 20 outputs to parallel to serial converter 72. Information from parallel to serial converter 72 is output either directly to data ring 38, or to PCL 60. From PCL 60, information is output to data ring 38. Transfer of information between parallel to serial converter 72, PCL 60, and data ring 38 is bi-directional.

Parallel to serial converter 72 is coupled directly to MPU 18 to receive display data. Similarly, horizontal loop size register 62, horizontal pointer register 64, horizontal pitch register 66, mode and attribute register 68, and status register 70 are all coupled directly to MPU 18.

Horizontal loop size register 62 stores the value representing the number of bits of the cascaded shift registers in data ring 38. The value should be (Sn X Sb) - 1 where Sn is the number of shifter registers cascaded together, and Sb is the width of the shift registers in bits. For segment drivers 32 and 34, the value would be (2 X 160)- 1 since the numbering begins with zero. Horizontal loop size register 62 controls the number shift clocks per operation.

Horizontal pointer register 64 latches two horizontal pointers which make up a boundary within the (Sn X Sb) - 1 loop. The pointers are called left and right pointers, but the pointers are interchangeable. The pointers define the area within the entire loop wherein MPU 18 will alter, amend, or replace display data. Display data within the data stream in data ring 38 that is not bounded by the two pointers (data within the pointers termed active region and data without termed inactive region) will remain unchanged by MPU 18.

Horizontal pitch register 66 dictates the width of string of bits that will be altered by MPU 18 within the active region of the entire data stream.

Mode and attribute register 68 directs the type of operation; that is whether the operation will be a sequential write or sequential read operation, as well as one-shot insert or replace. Mode and attribute register 68 also directs the attributes of the display screen. In other words, mode and attribute register 68 directs the type of background (white on black or black on white), controls superposition, highlights, etc.

Status register 70 contains the current status of row data serial interface 24, and registers several flags. A first flag, if set, indicates the data ring is currently halt data shifting and MPU 18 can access the shift registers. The second flag, if set, shows the current bit is within the active region. A third flag is set when an overflow has happened in horizontal pitch register 66. A fourth flag, when clear, indicates there is no operation currently carried out by row data serial interface 24.

Display data in RAM 48 is altered or replaced as follows.MPU 18 sets left and right pointers by writing values into horizontal pointer register 64. These register values are continuously compared with the number of bits shifted. Once the bit position marked by a pointer reaches the parallel to serial converter 72, PCL 60 changes paths accordingly to included the parallel to serial converter 72 into the loop. The loop does not stop after the number of cycles specified by horizontal pitch register 66 is reached. If a sequential operation (i.e. sequential read or sequential write) is selected, the loop will stop for MPU 18 access. If a one-shot replace is selected, the loop will not stop, only PCL 60 will switch paths to exclude the parallel to serial converter 72 from the loop. If a one-shot insert is selected, the loop will continue to advance and parallel to serial converter 72 will continue in the loop. Once the bit position marked by another pointer reaches parallel to serial converter 72, parallel to serial converter 72 will be excluded from the loop. Loop advance will stop since the bits marked by pointers will have returned to their original positions.

To explain a sequential operation in detail, Fig. 7 shows a graphical representation of a sequential write operation. In a), the data stream of one loop is shifted around through PCL 60. Further, the active region is defined by the left and right pointers as latched in horizontal pointer register 64. New information which will replace the display data in the active region is latched in parallel to serial converter 72. When the location of the left pointer reaches parallel to serial converter 72 (b), the link from the data ring 38 to PCL 60 is terminated and information within data ring 38 is connected directly to parallel to serial converter 72. Parallel to serial converter 72 is then coupled to PCL 60. In c), the location of the left pointer has been wrapped around through PCL 60 and the new information (W) has replaced the old (Z). At this time, the flow of data ring 38 halts to allow MPU 18 to write new information to parallel to serial converter 72. The procedure continues (d) until all the display data in the active region is altered as required and the active region returns to its original shift register 50 (e). A control signal then directs control logic 42 to fetch the altered display data from shift register 50 to RAM 48.

The pointers mentioned above are not limited to the system described. The pointers may be used with any system where serial write and serial read operations are performed on a data stream.

The above described LCD controller/driver 10 facilitates quick and easy alterations of rows of display data within segment drivers. Further, the pointers readily define the active region and facilitate quick and efficient data stream alteration.

Thus there has been provided, in accordance with the present invention, data stream pointers for a LCD support system that fully satisfies the objects, aims, and advantages set forth above. While the in-

vention has been described in conjunction with specific embodiments thereof, it is evident that many alternatives, modifications, and variations will be apparent to those skilled in the art in light of the foregoing description. Accordingly, it is intended to embrace all such alternatives, modifications, and variations as fall within the spirit and broad scope of the appended claims.

Claims

1. In a system where a string of data (38) may be used repeatedly, and portions of the data string may be altered by a central controller such as a micro-processing unit (MPU) (18), the data string coupled to a MPU interface device such as, but not limited to, a parallel-to-serial converter (72), and the data string further coupled to a logic circuit (60) for controlling the path of flow of the data string, the MPU interface device and the logic circuit for controlling the path coupled to a pointer register (64), a method of designating an area within the data string to be altered whereby other portions of the data string are ignored by the MPU, the method comprising the steps of:

   designating a beginning and an end of the location of the portion of the data string to be altered;

   storing in the pointer register a first value representing the beginning of the portion of the data string to be altered;

   storing in the pointer register a second value representing the end of the portion of the data string to be altered;

   storing a new bit of data in the MPU interface device;

   clocking the data string through the logic circuit until the location represented by said first value is reached;

   shifting the new bit of data from the MPU interface device into the data string;

   shifting a first bit of the data string out of the data string through the MPU interface device;

   clocking the data string another bit;

   storing another new bit of data in the MPU interface device;

   shifting the other new bit of data from the MPU interface device into the data string;

   shifting a second bit of the data string out of the data string through the MPU interface device; and

   repeating the above steps beginning from the step of clocking the data string another bit for subsequent bits until the location represented by said second value is reached.

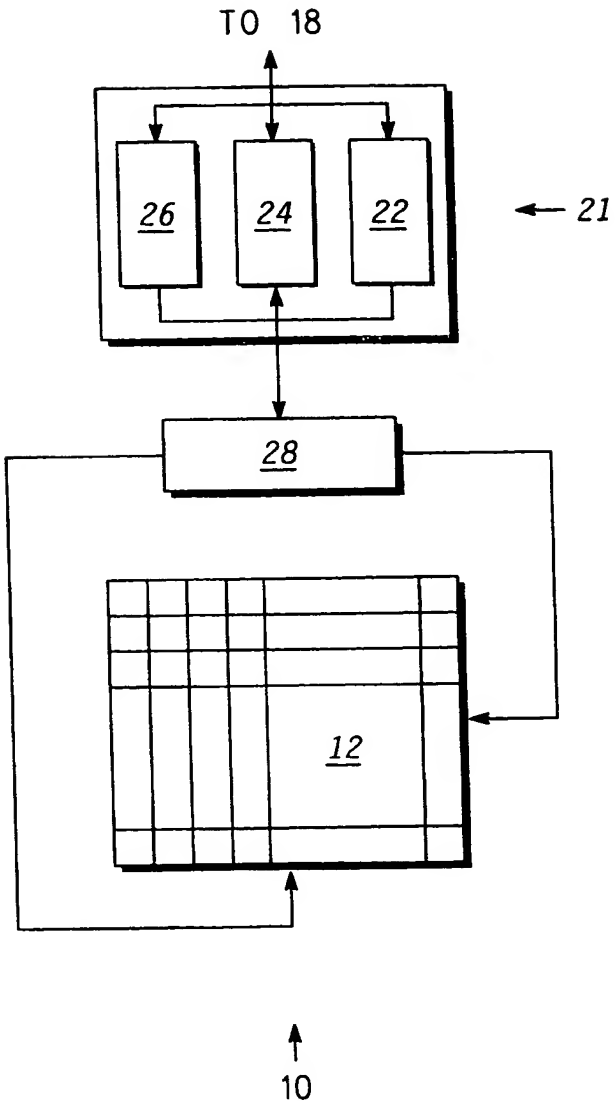


FIG. 1

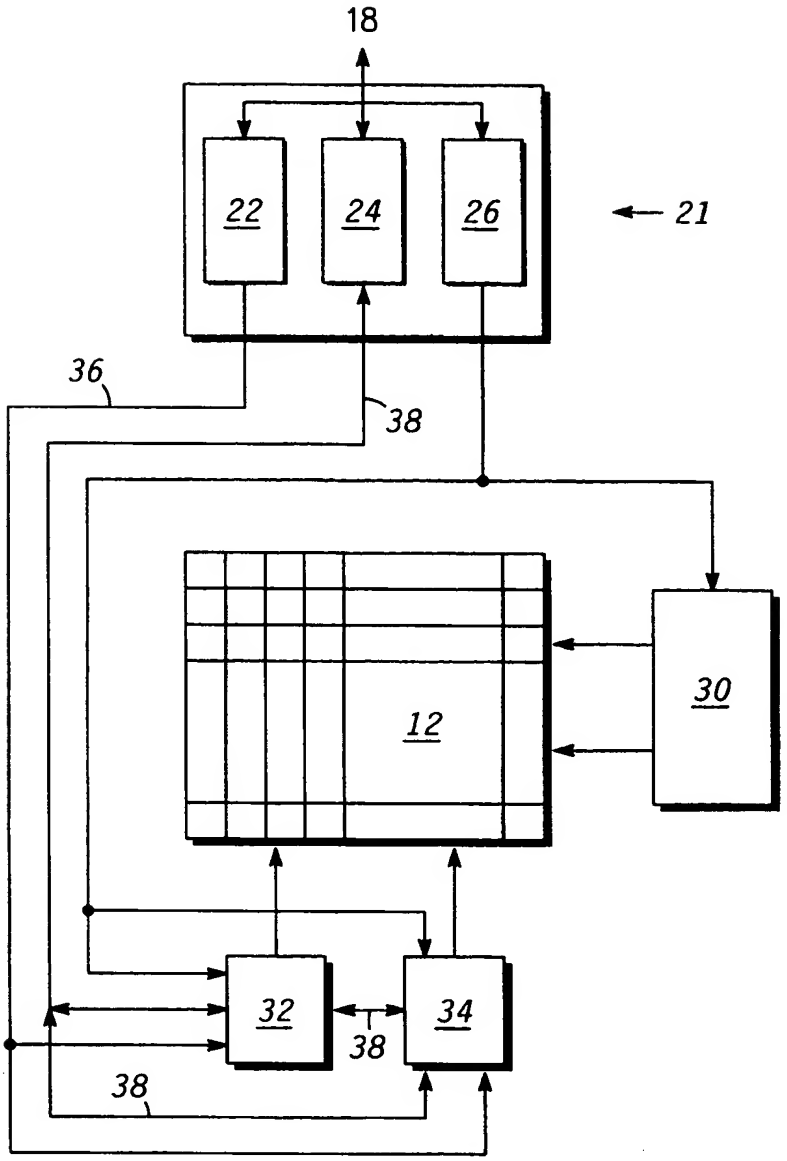


FIG. 2

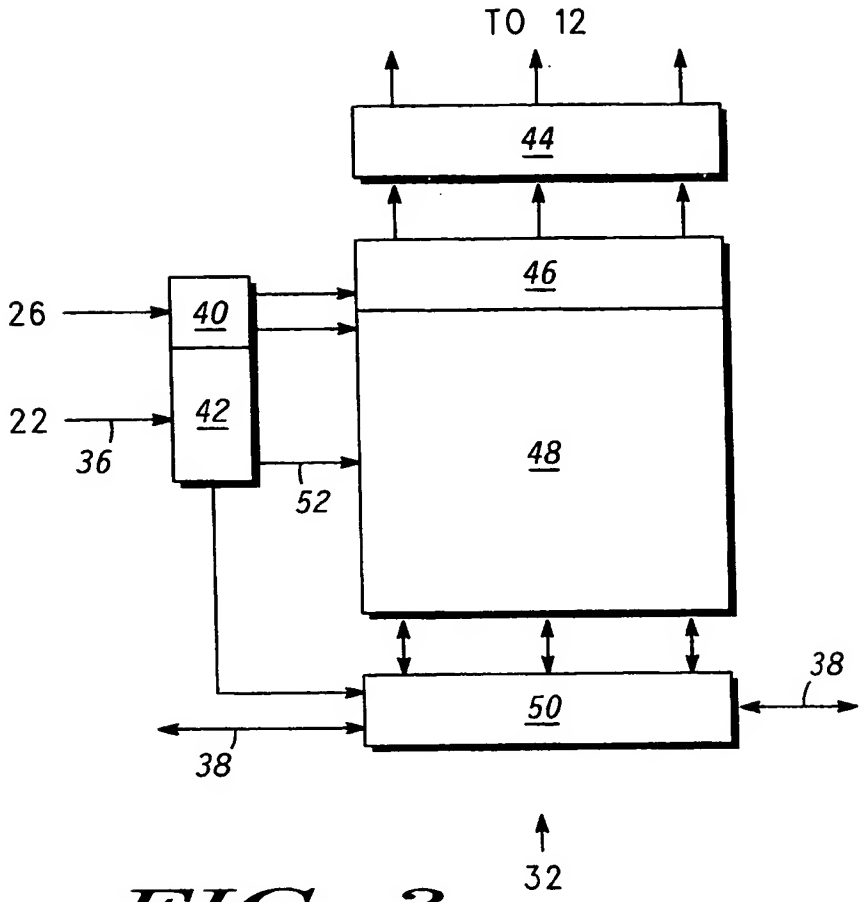
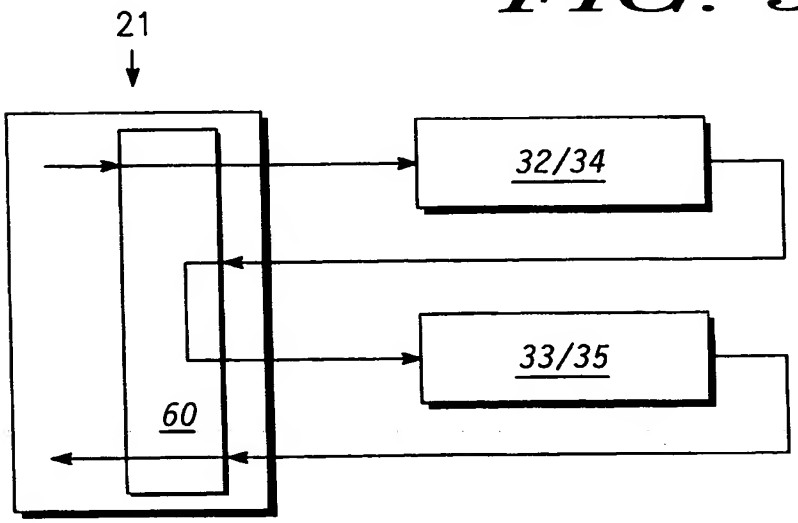


FIG. 3

FIG. 5



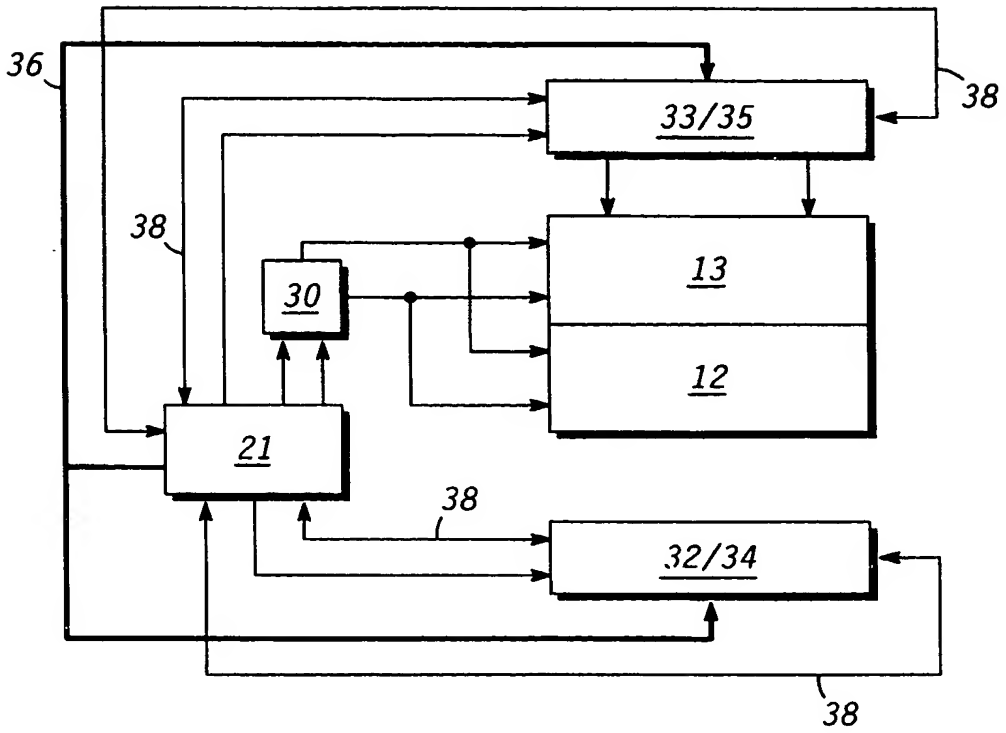
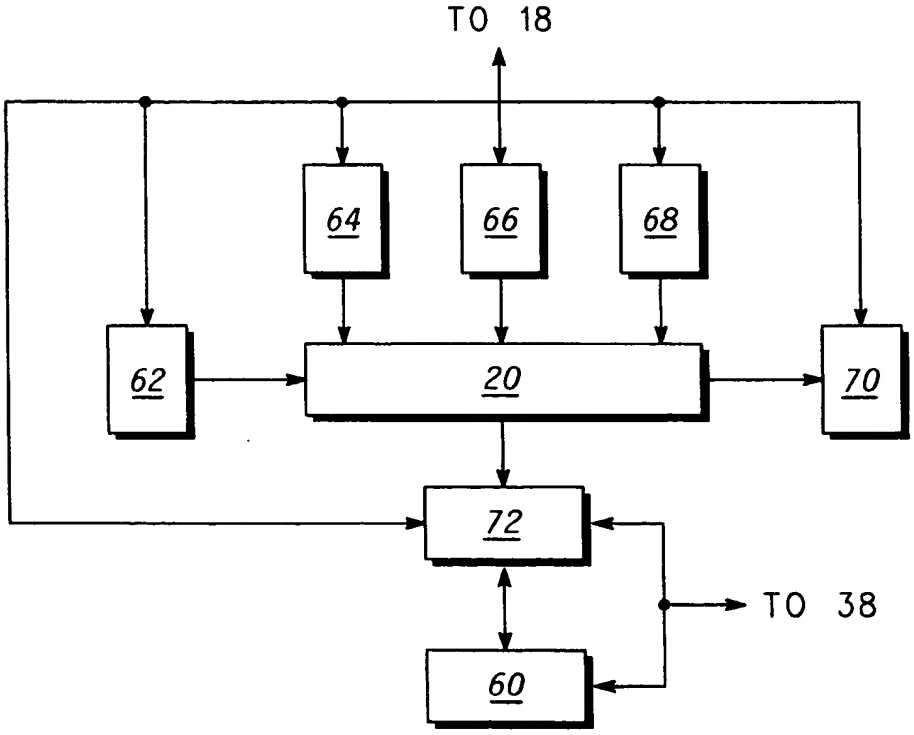


FIG. 4

FIG. 6



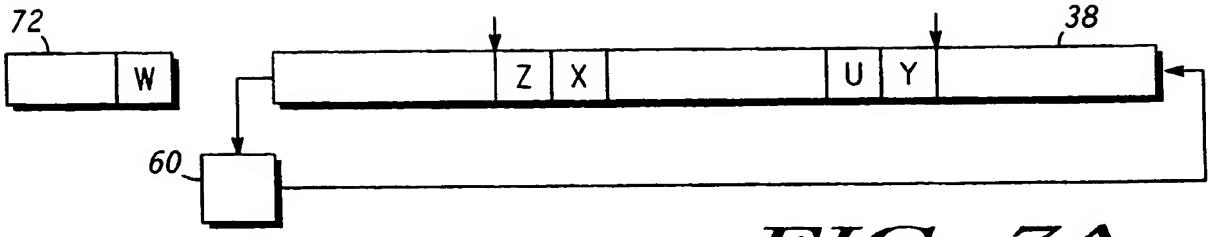


FIG. 7A

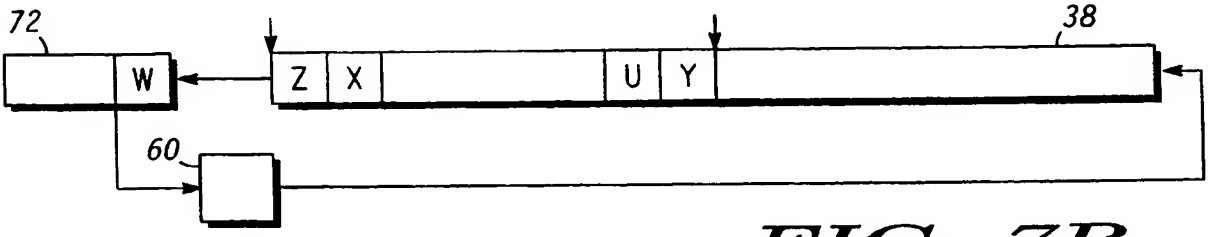


FIG. 7B

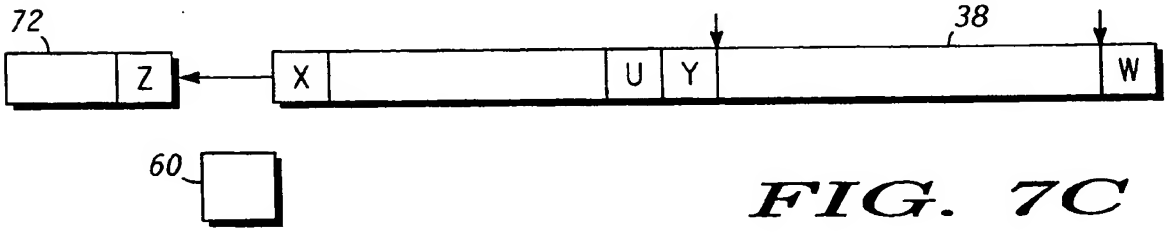


FIG. 7C

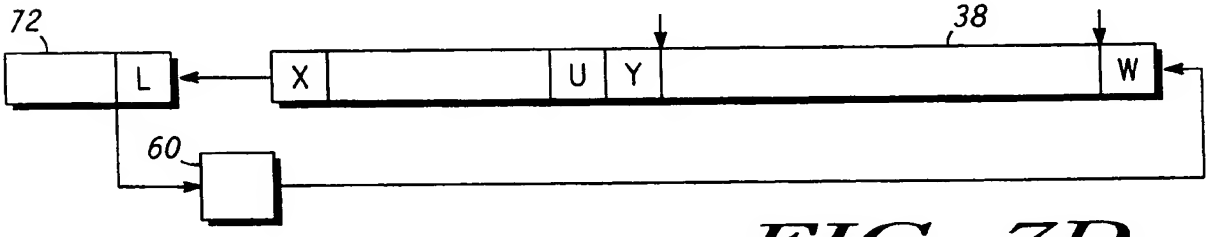


FIG. 7D

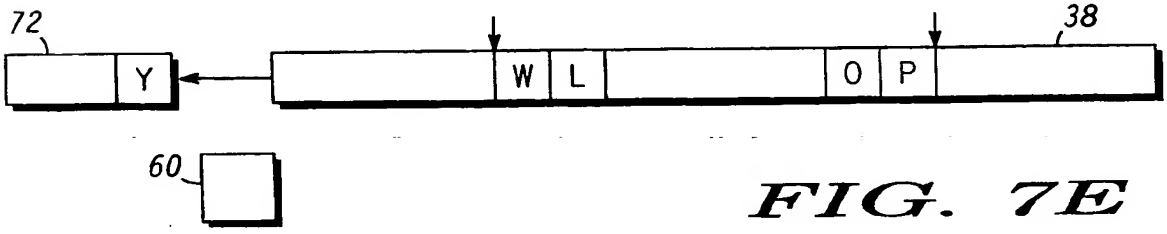


FIG. 7E